

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTERDISCIPLINAR**

**TECH BANK**

Serviços Financeiros Digitais

**Derek Gonçalves**

**Gabriel Oliveira**

**Geovani Monteiro**

**Kainan Guerrat**

**Kevin Nunes Amaral**

**São Paulo**

**2023**

**UNIVERSIDADE CRUZEIRO DO SUL CAMPOS SÃO MIGUEL**

**ANÁLISE E DESENVOLMENTO DE SISTEMAS**

**PROJETO INTERDISCIPLINAR**

**TECH BANK**

Serviços Financeiros Digitais

## Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.

**Orientadores:** Prof. Agnaldo Silibert Mota e Prof. Fabio Silva

**São Paulo**

**2023**

**SUMÁRIO**

[. 4](#_Toc133750878)

[1. INTRODUÇÃO 4](#_Toc133750879)

[1.1 Justificativa e Motivação 4](#_Toc133750880)

[1.2 Contextualização 4](#_Toc133750881)

[1.2.1 Plano de Negócios 5](#_Toc133750882)

[1.2.2 Regras de Negócios 6](#_Toc133750883)

[2 Requisitos de ENGENHARIA DE SOFTWARE 7](#_Toc133750884)

[2.1 Metodologia do Projeto 7](#_Toc133750885)

[2.2 Estrutura do Projeto 7](#_Toc133750886)

[2.2.1 Back-Log 9](#_Toc133750887)

[2.3 Requisitos do Sistema 13](#_Toc133750888)

[2.3.1 Requisitos Funcionais e Não Funcionais 13](#_Toc133750889)

[2.3.2 Requisitos Definidos 14](#_Toc133750890)

[2.3.3 Incertezas 15](#_Toc133750891)

[2.4 Análise SWOT 16](#_Toc133750892)

[3 Requisitos de ANÁLISE E DESENVOLVIMENTO DE PROJETO 17](#_Toc133750893)

[3.1 Desenvolvimento 17](#_Toc133750894)

[3.2 Modelo de Processo 17](#_Toc133750895)

[3.2.1 Fluxograma do Sistema 17](#_Toc133750896)

[3.2.2 Diagrama de Classes 17](#_Toc133750897)

[3.3 Casos de Uso 17](#_Toc133750898)

[3.3.1 Modelo de Casos de Uso 18](#_Toc133750899)

[3.3.2 Diagrama de Casos de Uso 20](#_Toc133750900)

[3.4 Modelo de Classes 22](#_Toc133750901)

[3.4.1 Estrutura do Banco de Dados 22](#_Toc133750902)

[3.4.2 Diagrama de Fluxo de Dados 22](#_Toc133750903)

[4 Considerações finais 23](#_Toc133750904)

[4.1 Lições Aprendidas 23](#_Toc133750905)

[4.2 Conclusão 23](#_Toc133750906)

[5 REFERêNCIAS 24](#_Toc133750907)

# INTRODUÇÃO

O mercado financeiro bancário durante muito tempo foi dominado por grandes marcas que tornava a livre competição quase que inexistente. Corroborado pela educação financeira que o povo carece e falta de concorrência, muitas dessas empresas sempre prestaram um serviço de qualidade duvidosa, cobrando taxas altíssimas e fornecendo uma assistência ineficaz para seus clientes.

Com a constante evolução da tecnologia, a criação de bancos digitais tem se tornado cada vez mais uma tendencia de mercado. Desde 2016, percebesse uma crescente tomada de porcentagem deste mercado por “Fintechs” ou bancos digitais, O desenvolvimento de um banco digital é uma tendência crescente no setor financeiro, proporcionando aos clientes uma experiência de uso mais rápida, segura e conveniente. Para se obter uma estrutura que atenda às necessidades do cliente é fundamental que o desenvolvimento seja realizado de forma estruturada e planejada, utilizando conceitos de Engenharia da Software e Análise e Desenvolvimento de Sistemas, afim de garantir a qualidade e a segurança desses serviços.

## Justificativa e Motivação

O objetivo deste trabalho é apresentar uma documentação completa e detalhada para o desenvolvimento de um banco digital, demonstrando o domínio dos conceitos de Engenharia de Requisitos e dos modelos de gestão de projeto. Através da definição de requisitos, especificações funcionais e não-funcionais, modelagem de processos de negócios, descrição da arquitetura do sistema e definição de casos de uso e fluxos de trabalho, será possível garantir um desenvolvimento eficiente e de qualidade.

Por meio da aplicação desses conceitos e modelos, espera-se entregar uma documentação completa e coerente para o desenvolvimento de um banco digital, que atenda às expectativas do cliente e dos usuários finais.

## Contextualização

O grupo definiu de maneira democrática, a criação de uma empresa fictícia para basearmos o desenvolvimento da proposta do projeto, o nome dado foi “Tech Bank”. “Tech Bank” é um banco de serviços financeiros digitais que oferece soluções inovadoras para clientes em todo o mundo. É fornecido serviços bancários online seguros e convenientes, incluindo contas correntes, cartões de crédito e débito, empréstimos e muito mais. Com a tecnologia mais avançada e uma equipe de especialistas em finanças, ela se compromete a fornecer aos seus clientes a melhor experiência bancária possível.

## Plano de Negócios

Tech Bank é um banco de serviços financeiros digitais que oferece soluções inovadoras para clientes em todo o mundo. Nós fornecemos serviços bancários online seguros e convenientes, incluindo contas correntes, cartões de crédito e débito, empréstimos e muito mais. Com a tecnologia mais avançada e uma equipe de especialistas em finanças, estamos comprometidos em fornecer aos nossos clientes a melhor experiência bancária possível. O Tech Bank é a escolha ideal para quem procura um banco moderno e eficiente, com todas as vantagens do mundo digital.

* Missão: Nós, do Tech Bank, buscamos oferecer a melhor experiência bancária possível, valorizando a transparência, confiabilidade e excelência em tudo o que fazemos. Nossa missão é ajudar nossos clientes a alcançar seus objetivos financeiros e ser uma força positiva na comunidade.
* Valores:

1. Segurança - Temos o dever de proteger e assegurar a segurança de nossos colaboradores.

2. Velocidade - Entregamos a melhor performance e velocidade em suas transações e solicitações.

3. Acessibilidade - Damos total suporte a todas as pessoas e empresas com nossa simplicidade e fácil acesso.

* Produtos e Serviços:

Oferecemos soluções de negócios personalizados para empresas de todos os portes, incluindo contas corporativas, serviços de pagamento, financiamento de projetos e muito mais. Nós nos concentramos em atender às necessidades financeiras de nossos clientes de maneira rápida, segura e eficiente. Observe que o Tech Bank não oferece serviços de investimento.

* Estratégia De Marketing:

Queremos oferecer soluções financeiras inovadoras e personalizadas para atender às necessidades individuais de nossos clientes. Para alcançar isso, usamos mídias sociais e outras plataformas digitais para conectar com nossos clientes em potencial e fornecer uma experiência de marketing envolvente e interativa. Priorizamos a transparência e a simplicidade em todas as nossas comunicações com o consumidor, tornando nossos produtos e serviços fáceis de entender e acessíveis para todos. Além disso, nos dedicamos para fornecer um atendimento ao cliente de alta qualidade em todas as confortáveis e soluções rápidas e eficazes para problemas que possam surgir, e assim crescemos mais e mais através de indicações dos nossos clientes fiéis.

## Regras de Negócios

Tendo por definido o “Big Hairy Audacious Goals” como sendo se tornar o prestador de serviços bancários digitais mais influente da América Latina, existem algumas diretrizes que devem ser seguidas, para tornar o objetivo descrito acima uma realidade, desde a parte de desenvolvimento do produto quanto ao contato com o cliente final no pré e pós contato.

- Regras relativas a conta bancária:

* A conta bancária só pode ser aberta com informações precisas e completas do cliente, incluindo nome, endereço, número de identidade e informações de contato.
* O cliente pode solicitar a abertura de uma conta corrente ou de poupança.
* O banco deve fornecer serviços bancários 24 horas por dia, 7 dias por semana.
* O cliente pode realizar transferências eletrônicas, pagamentos de contas e outras transações financeiras através do sistema online do banco.
* O banco deve garantir a segurança das transações financeiras e dos dados do cliente, utilizando técnicas de criptografia e autenticação adequadas.

- Regras relativas ao contato empresa-consumidor:

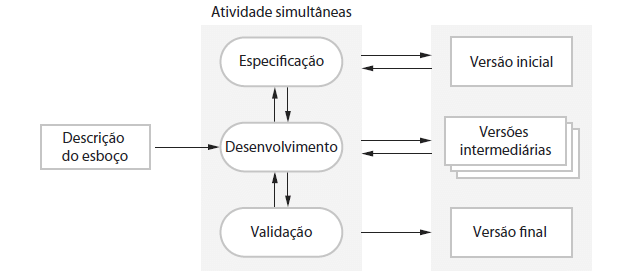
* O banco deve verificar a identidade do cliente antes de aprovar a abertura da conta.
* O banco deve fornecer ao cliente informações claras e precisas sobre as taxas, encargos e benefícios de cada tipo de conta oferecida.
* O banco deve fornecer atendimento ao cliente eficiente e amigável, através de canais online, móveis ou por telefone.
* O banco pode oferecer serviços adicionais, como empréstimos pessoais ou cartões de crédito, desde que os requisitos e termos sejam claros para o cliente.

# 2 Requisitos de ENGENHARIA DE SOFTWARE

## 2.1 Metodologia do Projeto

A metodologia utilizada para o desenvolvimento deste projeto foi a Iterativa ou Evolucionária: a mesma é capaz de combinar aspectos tanto do método cascata quando da metodologia ágil, permitindo uma abordagem mais flexível e adaptável ao longo do desenvolvimento. Ela se baseia em ciclos de desenvolvimento curtos, como nos métodos ágeis, mas com a possibilidade de revisão e mudança de requisitos em cada ciclo, como na cascata. No caso do desenvolvimento de um banco digital, a metodologia iterativa é uma opção mais adequada, pois permite uma maior flexibilidade para adaptar-se às mudanças no mercado financeiro e nas demandas dos clientes. Além disso, a metodologia iterativa permite uma maior agilidade no desenvolvimento, pois as equipes podem trabalhar em paralelo em diferentes partes do sistema, sem a necessidade de finalizar completamente uma etapa para iniciar outra. Isso pode acelerar o desenvolvimento e possibilitar que o produto final seja entregue em um prazo mais curto.

Com isso, denota-se que tal metodologia se adapta de forma melhor a proposta do projeto, assim fica a critério do P.O do projeto, demandar a entrega das etapas dentro dos prazos favoráveis.

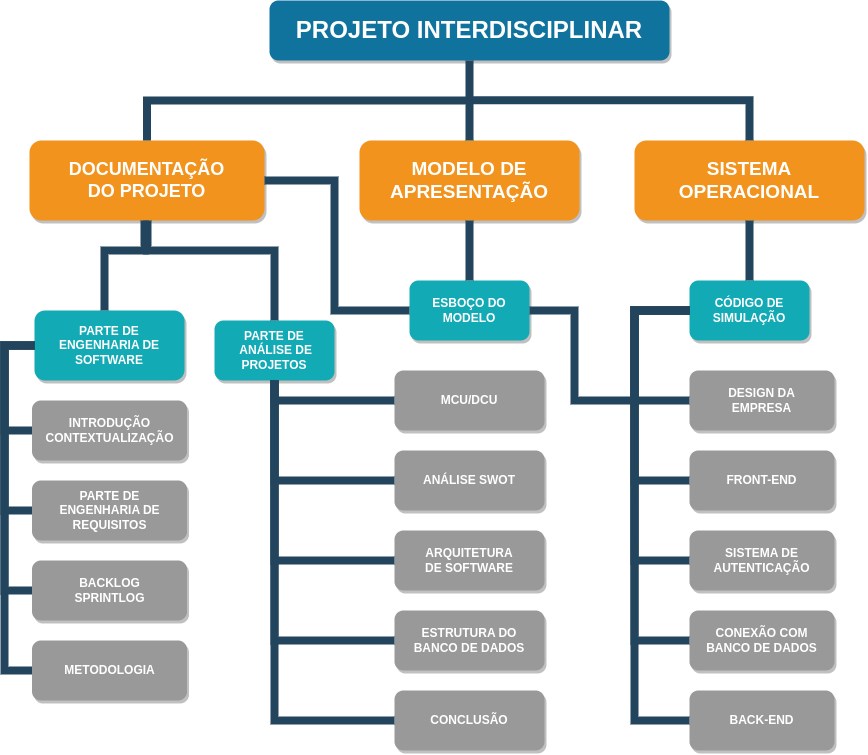


## 2.2 Estrutura do Projeto

Para início do projeto, o grupo foi dividido em 3 frentes de desenvolvimento:

* + Documentação: Kainan Guerra e Kevin Nunes
  + Sistema Operacional: Geovane Monteiro e Gabriel Oliveira
  + Slides para Apresentação: Derek Gonçalves

Cada qual atribuição de tarefa foi pensada nas competências de cada um, havendo recolocações no decorrer do projeto. Desta forma, cada time poderia trabalhar de forma independente, sendo necessário apenas a comunicação em alguns pontos de cada processo. Por exemplo: para desenvolvimento dos slides, o responsável precisaria da parte de design do projeto do sistema (Front-End), a fim de combinar paleta de cores, para entregar um modelo de apresentação compatível com a imagem adotada para a empresa fictícia criada. O time de desenvolvimento do sistema, por sua vez, necessitaria da parte de engenharia de requisitos para que possa construir o código de maneira fidedigna a elaborada pelo time de documentação.



## 2.2.1 Back-Log

Épico 1 (Semana 1) - Nesta primeira etapa, o foco é a definição do tema do projeto, tal como a abordagem que será utilizada e as atribuições de funções de cada integrante, desenvolvendo assim, as primeiras tarefas que vão dar o escopo e arcabouço do projeto como um todo. Não obstante, saber identificar potenciais talentos para cada função se faz essencial para o manejamento da equipe, visando a entrega no prazo determinado, mitigando potenciais atrasos, contratempos ou situações adversas.

Épico 2 (Semana 2) - Tendo as funções definidas e as primeiras tarefas concluídas, tem-se como base uma visão de progresso de trabalho de todas as partes do grupo (quais são as competências de cada um, como cada um se empenha individualmente e pontualidade dos mesmos). Sendo assim, está etapa serve para avaliar se o desenvolvimento está ou não convergindo para o resultado final estimado.

Se sim, novas tarefas são formuladas e passadas para os integrantes.

Se não, reatribuições ou recolocações devem ser consideradas.

Épico 3 (Semana 3) - Dando prosseguimento no projeto, talvez sejam encontradas dificuldades que impeçam o prosseguimento de uma tarefa específica, o que por sua vez impede a conclusão do projeto, com isso, é de suma importância que o P.O ou responsável saiba do problema o quanto antes para conseguir solucioná-lo antes que se torne algo impossível de se resolver antes do prazo de entrega (efeito bola de neve). Espera-se que nesta etapa, ao menos 50% do trabalho esteja formado.

Épico 4 (Semana 4) – Neste momento, a parte que possui a prioridade mais alta estará concluída (neste caso, refere-se ao código do sistema que simula as ações do usuário). A partir daqui, entende-se que nada impede o prosseguimento da parte escrita e desenvolvimento dos slides da apresentação. Pesquisas bibliográficas devem já estar em andamento demostrando resultados nítidos de competência no que se refere as regras da ABNT. Uma pessoa há de ficar responsável pela construção dos slides, logo, a comunicação do mesmo com os outros do time deve ser constante.

Finalização – Em um cenário ideal, o projeto estará completo antes do limite de entrega, faltando algo em torno de 2 semanas para a data de apresentação, dando uma margem de segurança para Enhancements (“melhorias”) que sirvam para incrementar o projeto, um tempo que também pode ser aproveitado para organização das falas de cada integrante e ensaios das mesmas. Caso contrário, é avaliado cada processo restante, priorizando o desenvolvimento do mesmo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STATUS** | **TAREFA** | **DESCRIÇÃO** | **INÍCIO** | **FIM** |
| Done | TPI-01 | *A fim de incrementar e iniciar a parte escrita do projeto, é necessário colocar por escrito as definições do que são e as regras de negócios empresa em si.* | 29/03/23 | 15/04/23 |
| Done | TPI-02 | *Após a parte de introdução da parte escrita, irá aparecer um breve plano de negócios da empresa, com isso, é preciso desenvolvê-lo por escrito.* | 29/03/23 | 03/04/23 |
| Done | TPI-03 | *Um dos critérios de aceite do projeto interdisciplinar, é o desenvolvimento de um código que simule os requisitos do sistema.* | 01/04/23 | 07/05/23 |
| Done | TPI-04 | *A imagem da empresa é um dos componentes principais de qualquer negócio, com isso, é preciso desenvolver logo, slogan e definir a paleta de cores que será utilizada.* | 30/03/23 | 04/04/23 |
| Done | TPI-05 | *Para se ter um arcabouço do projeto, é interessante já termos um esboço do que iremos escrever. Desenvolver a estrutura em ABNT para deixar para futuras inserções.* | 02/04/23 | 16/04/23 |
| Done | TPI-06 | *A fim de se ter uma base de desenvolvimento do código, precisa-se montar telas com componentes essenciais para se basear o desenvolvimento da parte do código.* | 02/04/23 | 22/04/23 |
| Done | TPI-07 | *Para incorporação da parte escrita, precisa-se ter uma base sólida de referências da área de engenharia de software e análise de projeto.* | 01/04/23 | 15/04/23 |
| Done | TPI-08 | *Toda base do back-end é feita em referência ao modelo de banco de dados utilizado no sistema. Com isso, definir as relações de tal componente é de suma importância.* | 03/04/23 | 09/04/23 |
| Done | TPI-09 | *Em referência a estrutura da parte escrita do projeto, é preciso começar a estruturar os slides da apresentação que será feita, utilize PowerPoint ou algum outro software para tal. A criatividade é incentivada nesta tarefa. Competência e progresso serão avaliados.* | 03/04/23 | 16/04/23 |
| Done | TPI-10 | *Uma das partes essenciais do projeto, principalmente da parte escrita, é a formulação de um back-log que envolva todas as partes de construção do projeto.* | 01/04/23 | 09/04/23 |
| Done | TPI-11 | *A parte escrita do projeto será composta por alguns fluxogramas e digramas UML que podemos ver na aula de análise, com isso, desenvolver um que demonstre como ocorre a ação do usuário é de suma importância para incrementar o projeto.* | 10/04/23 | 15/04/23 |
| Done | TPI-12 | *Fazer um fluxograma que exponha como ocorre o fluxo dos dados dentro do sistema se faz indispensável para o desenvolvimento do projeto, tendo em vista que vimos isso na matéria de Análise de Projeto de Sistemas.* | 10/04/23 | 22/04/23 |
| Done | TPI-13 | *O objetivo desta tarefa é definir o ciclo de vida e a metodologia de desenvolvimento do projeto de forma clara e precisa, considerando as necessidades e requisitos do cliente, bem como as melhores práticas da indústria de engenharia de software.* | 21/04/23 | 31/04/23 |
| Done | TPI-14 | *Escrever uma análise SWOT do sistema para identificar os pontos fortes, pontos fracos, oportunidades e ameaças do projeto, permitindo que sejam tomadas decisões informadas e estratégias de mitigação de riscos sejam definidas.* | 22/04/23 | 07/05/23 |
| Done | TPI-15 | *Identificar, analisar e documentar os requisitos funcionais e não funcionais do sistema, garantindo que as necessidades e expectativas do cliente e de outros stakeholders sejam atendidas, bem como os padrões de qualidade e desempenho sejam alcançados.* | 01/04/23 | 12/04/23 |
| Done | TPI-16 | *Documentar as lições aprendidas durante todo o ciclo de vida do projeto, incluindo o que funcionou bem e o que poderia ter sido melhorado. Além disso, a conclusão do projeto deve ser elaborada, resumindo o sucesso do projeto como um todo.* | 16/04/23 | 14/05/23 |
| Done | TPI-17 | *Criar um fluxograma detalhado do sistema que ilustre o fluxo de dados e a arquitetura aplicada para o projeto, incluindo a definição das interfaces, componentes e seus relacionamentos. Permitindo uma visão clara do sistema como um todo e facilitando a comunicação entre os membros da equipe de desenvolvimento.* | 28/03/23 | 13/04/23 |
| Done | TPI-18 | *Criar a estrutura analítica do projeto, que representa visualmente todas as entregas e atividades necessárias para alcançar os objetivos do projeto. A estrutura analítica deve ser desenvolvida de forma hierárquica, detalhando as entregas principais em tarefas menores e mais gerenciáveis, permitindo a definição de cronogramas, estimativas de custos e alocação de recursos.* | 24/03/23 | 28/04/23 |
| Done | TPI-19 | *Fazer o diagrama de classes de objetos, que representa visualmente as classes de objetos, seus atributos e métodos, e seus relacionamentos, permitindo uma melhor compreensão da estrutura do sistema.* | 22/04/23 | 29/04/23 |
| Done | TPI-20 | *Descrever a abordagem que será utilizada para gerenciar e executar o projeto, bem como a estrutura organizacional da equipe de desenvolvimento. A metodologia deve ser escolhida de acordo com as características do projeto, enquanto a estrutura da equipe deve ser planejada de acordo com as habilidades e competências necessárias para o sucesso do projeto.* | 01/04/23 | 04/04/23 |
| Done | TPI-21 | *Desenvolver a parte de autenticação do usuário, que permitirá a validação das credenciais do usuário e o acesso seguro ao sistema. Isso inclui a implementação das funcionalidades de cadastro de usuários, login e recuperação de senha, além de garantir a segurança e privacidade dos dados do usuário por meio de medidas de proteção adequadas, como criptografia e autenticação de dois fatores.* | 11/04/23 | 13/05/23 |

## 2.3 Requisitos do Sistema

Em engenharia de software, temos o que se denomina como engenharia de requisitos, uma parte essencial para o processo de desenvolvimento de sistemas, já que, os requisitos, tanto funcionais quanto não funcionais, regem a forma como o mesmo irá se comportar em determinadas situações quando o usuário interagir consigo.

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada.” (Sommerville, 2011. p 57)

### 2.3.1 Requisitos Funcionais e Não Funcionais

* Requisitos funcionais: Tudo aquilo que engloba as funcionalidades visíveis do sistema, as quais interagem diretamente com a ação do cliente, podem ser consideradas requisitos funcionais, já que, expressam com notoriedade a execução de uma função dentro do sistema de software. Um apertar de um botão que abre um pop-up ou um modal, um algoritmo que faça a autenticação de login do usuário são exemplos de requisitos funcionais pois são fundamentais para a execução e interação do programa com o cliente final.

“São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Os requisitos funcionais também podem explicar o que o sistema não deve fazer” (Sommerville, 2011. p 59)

* Requisitos não funcionais: São aqueles requisitos que dizem respeito ao ambiente externo da organização, normalmente uma norma governamental ou até mesmo uma restrição que o próprio sistema deve possuir, algo especifico que não impede o funcionamento geral do mesmo, mas de acordo com os critérios do cliente se faz necessário para atender alguma demanda, seja uma regra de negócio ou lei de diretriz daquele seguimento de mercado.

“São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas por normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.” (Sommerville, 2011. p 59)

### 2.3.2 Requisitos Definidos

- O sistema inicialmente apresenta uma página de login com:

• dois campos para inserção de texto(e-mail e senha)

• um botão de acessar que só prossegue após a validação dos campos e-mail e senha

• um botão de cadastrar para caso o cliente não tenha acesso. Ao ser clicado direciona o usuário para um formulário onde os dados devem ser preenchidos e clicado em um botão de enviar ou "criar conta"

• deve existir um botão de esqueci minha senha que abre um modal onde pode ser inserido o e-mail do usuário

- Após a validação, o usuário acessa a home do sistema, onde ele tem acesso a:

• um botão que leva o mesmo a fazer transferências bancárias

• um botão que o leva a emitir uma ordem de depósito

• um botão que o leva a emitir uma ordem de saque

• uma aba lateral com seus dados bancários, saldo e um botão que o direciona a outra guia de cartões

• na aba lateral existe um botão de histórico de operações onde ao ser clicado, é levado a uma página onde se pode ver todas as ordens emitidas, tanto de transferências, quanto de depósito ou saque, feitas pelo cliente

- A parte de transferências é um formulário de emissão de ordens que:

• É possível fazer transferências entre contas do próprio banco ou outros bancos

• Ao ser preenchido todos os dados do formulário, uma nova ordem de transferência é criada e um modal aparece dizendo: "nova ordem criada com sucesso: n° xxxxxxxx" juntamente com o número da ordem.

• O cliente pode verificar o sucesso da transação conferindo em seu saldo bancário na página home.

- Na página de depósito:

• O cliente pode criar uma ordem de depósito preenchendo seus dados de origem daquela quantia e o valor em si

• Após a verificação ou match das informações dentro da conta do próprio banco, pressupondo que o valor já foi depositado, o saldo é adicionado na conta do cliente e deduzido da conta principal do banco.

• Caso o valor não tenha sido depositado ainda, a ordem fica no status "awaiting\_payment".

- Na parte de saque do sistema:

• O cliente pode emitir uma ordem de saque ao preencher os campos obrigatórios

• O valor da ordem é deduzido do saldo do cliente

- Após o botão cartões ser clicado:

• Uma página com os cartões já existentes do cliente aparece

• Caso o mesmo não possua nenhum, aparece um botão "pedir novo cartão"

• Após ser preenchido as informações, selecionado o tipo do cartão e clicado em "submit", um modal confirma a ação bem sucedida do usuário

• O novo cartão emitido aparece na aba cartões mostrando tipo, validade e etc

### 2.3.3 Incertezas

Em Engenharia de Requisitos, as incertezas referem-se a uma falta de clareza ou precisão na compreensão dos requisitos do sistema. As incertezas podem surgir de várias fontes, incluindo a falta de conhecimento do domínio do problema, requisitos ambíguos, falta de informações ou imprecisão nas informações fornecidas pelo cliente ou usuário final.

As incertezas podem ter um impacto significativo na qualidade do produto final e no sucesso do projeto de desenvolvimento de software. Se não forem gerenciadas adequadamente, as incertezas podem levar a atrasos no projeto, custos adicionais, retrabalho e até mesmo a falha do projeto.

O objetivo deve ser reduzir as incertezas e garantir que os requisitos do sistema sejam compreendidos de forma clara e precisa, para que possam ser implementados corretamente e atender às necessidades do cliente ou usuário final.

Alguns exemplos das incertezas identificadas durante o processo de desenvolvimento foram:

Em relação ao software não está explicito os seguintes fatores:

• Somente 1 conta por CPF e CNPJ.

• Não é possível acesso simultâneo a mesma conta.

• Validação de usuário tem que ser feita pra qualquer tipo de movimentação, solicitação ou alteração de dado cadastral.

Em relação as operações de saque, não foi mencionado os seguintes parâmetros:

• Horário para efetuar saques das 8h da manhã até 20h da noite.

• Limite de saques por dia 1 até 2000 mil reais para pessoa física acima disso ligar na agência por questão de segurança.

• Limite de saques por dia 5 até 10 mil reais para pessoa jurídica acima disso ligar na agência por questão de segurança.

• Somente o legitimo proprietário da conta pode efetuar saques.

• Sem taxa de saque.

-Deposito:

• Horário para efetuar saques das 8h da manhã até 20h da noite.

• Sem limite de valor para depósitos tanto PF ou PJ.

• Sem taxa de deposito.

-Transferência:

• DOC e TED segue à risca as regras do banco central.

• Valor limite de transferências via pix por dia pode ser moldado pelo usuário.

• 24h de serviço para transferências via pix.

• DOC e TED somente 3 transferências por dia para pessoas físicas acima disso cobrada taxa de 5 reais ao usuário.

• DOC e TED somente 7 transferências por dia para pessoas jurídicas acima disso cobrada taxa de 15 reais ao usuário.

-Cartões:

• Usuários de pessoa física podem ter até no máximo 5 cartões.

• Usuários de pessoa jurídica podem ter até no máximo 10 cartões.

• Obrigatório possuir 1 cartão que possua função débito.

• Limite máximo de crédito para pessoa física e de no mínimo 500 reais a 20 mil reais, dependendo da renda mensal do usuário para poder solicitar um cartão.

• Renda mínima e de 1.200 reais para pessoa física.

• Limite máximo de crédito para pessoa jurídica e de no mínimo 10 mil reais a 500 mil reais, valor de crédito depende do faturamento semestral do CNPJ.

• Atualização de crédito semestral, tanto PF quanto PJ.

• Horário para solicitação de cartões e de 24h por dia de segunda a sábado menos aos domingos

## 2.4 Análise SWOT

A análise SWOT é uma ferramenta poderosa para avaliar a posição competitiva de uma empresa em seu mercado. No caso do Tech Bank, uma instituição financeira digital que oferece soluções personalizadas para seus clientes, a análise SWOT pode fornecer uma compreensão aprofundada de suas forças, fraquezas, oportunidades e ameaças.

Pontos Forte:

* Modelo de negócio totalmente digital, que permite maior eficiência operacional e agilidade no lançamento de novos produtos e serviços.
* Foco em soluções financeiras personalizadas para atender às necessidades individuais de cada cliente.
* Tecnologia avançada e segurança cibernética para garantir a privacidade e a segurança dos clientes.

Pontos Fraco:

* Falta de presença física pode limitar a confiança dos clientes em potencial em relação ao banco.
* Dependência de fornecedores terceirizados para fornecer suporte técnico, telemarketing e de infraestrutura.

Oportunidades:

* Crescimento do mercado de serviços financeiros digitais em todo o mundo, oferecendo oportunidades de expansão.
* Avanços contínuos em tecnologia, oferecendo oportunidades para o Tech Bank expandir seus serviços e oferecer soluções ainda mais inovadoras.

Ameaças:

* Concorrência acirrada entre bancos digitais e tradicionais.
* Regulamentos e regras em constante mudança pode afetar o desenvolvimento de novos produtos em diferentes áreas.
* Ameaças cibernéticas em constante evolução pode afetar a confiabilidade do nosso banco com nossos clientes.

# 3 Requisitos de ANÁLISE E DESENVOLVIMENTO DE PROJETO

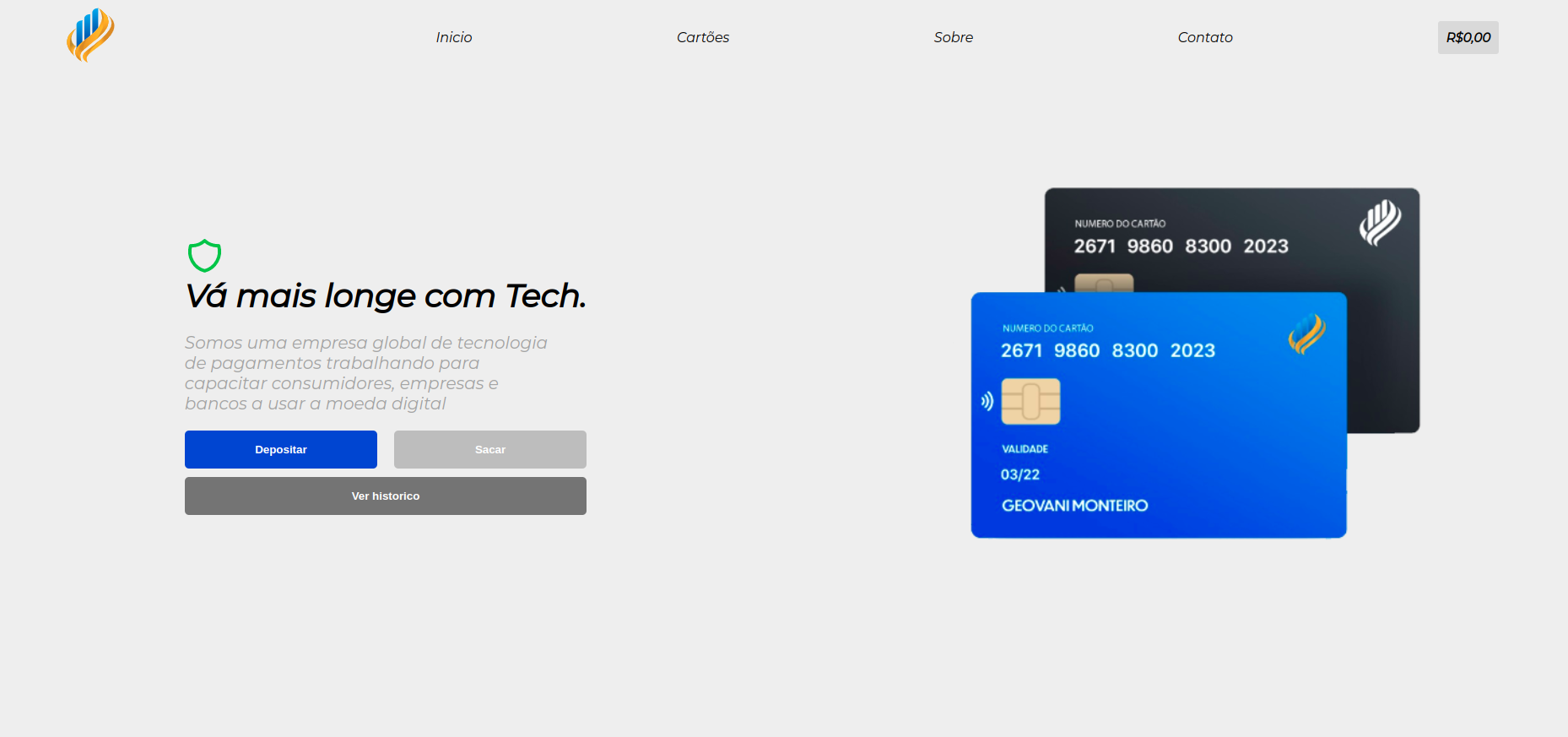
Esta parte do projeto é dedicada a disciplina de Análise e Desenvolvimento de Sistemas, buscando evidenciar suas competências e conteúdos aprendidos em sala, no que se refere as ferramentas e técnicas utilizadas para aprimoramento das etapas de construção de softwares.

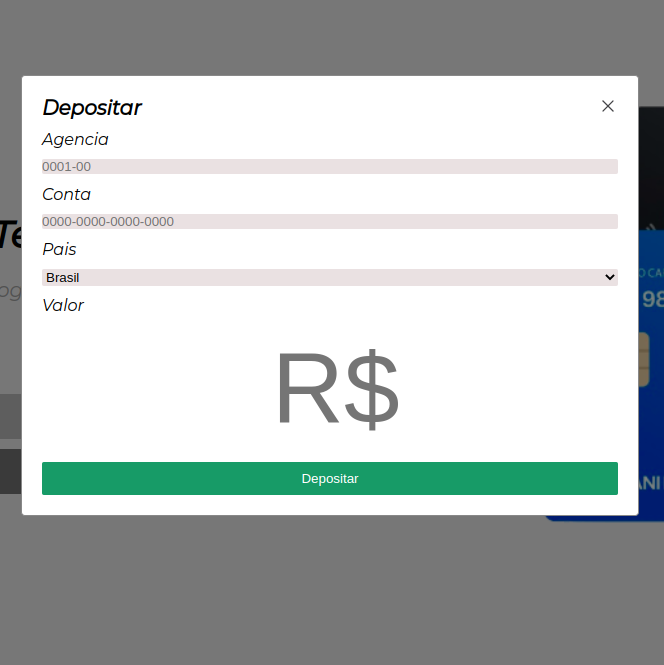
## 3.1 Desenvolvimento

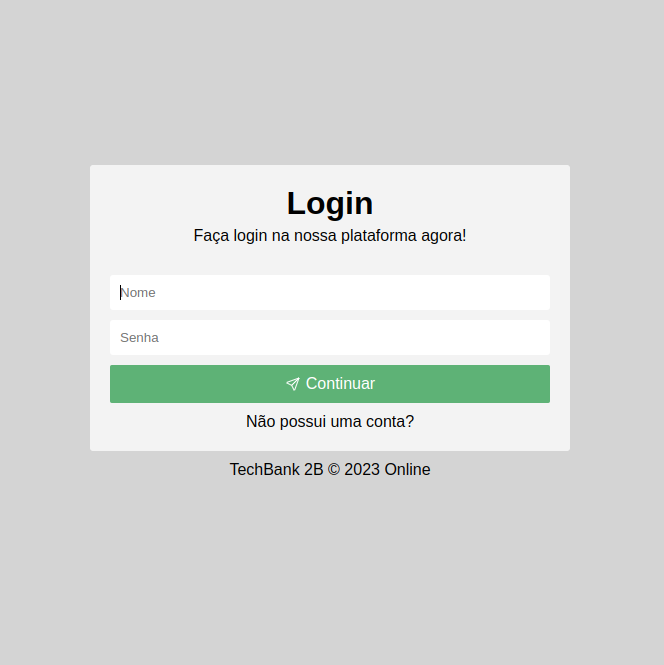
* UI/UX DESIGN:



* INTERFACE:





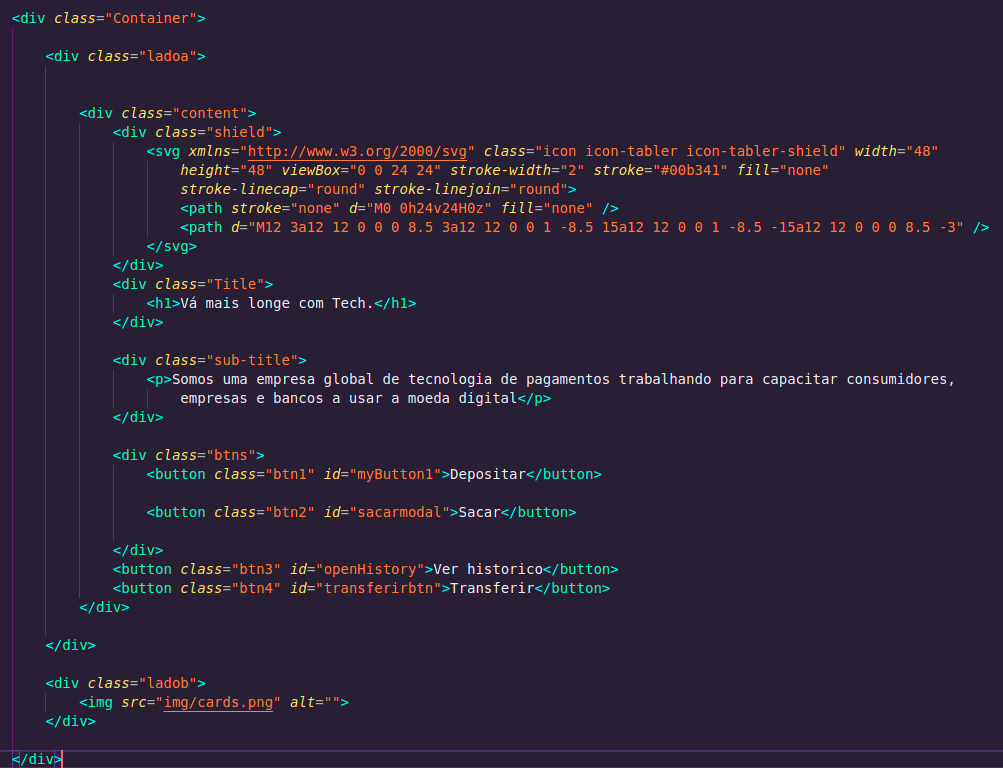


* CÓDIGO:

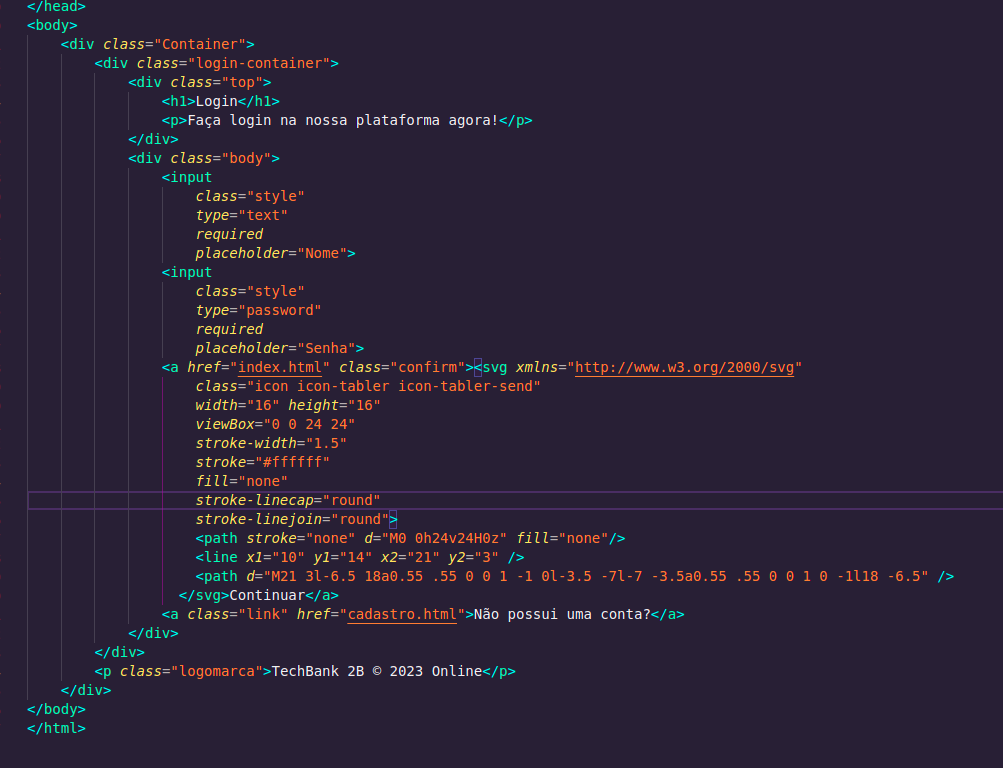


- Barra de Navegação: ->

- Página Principal: ->



- Página de Login: →



## 3.2 Arquitetura de Software

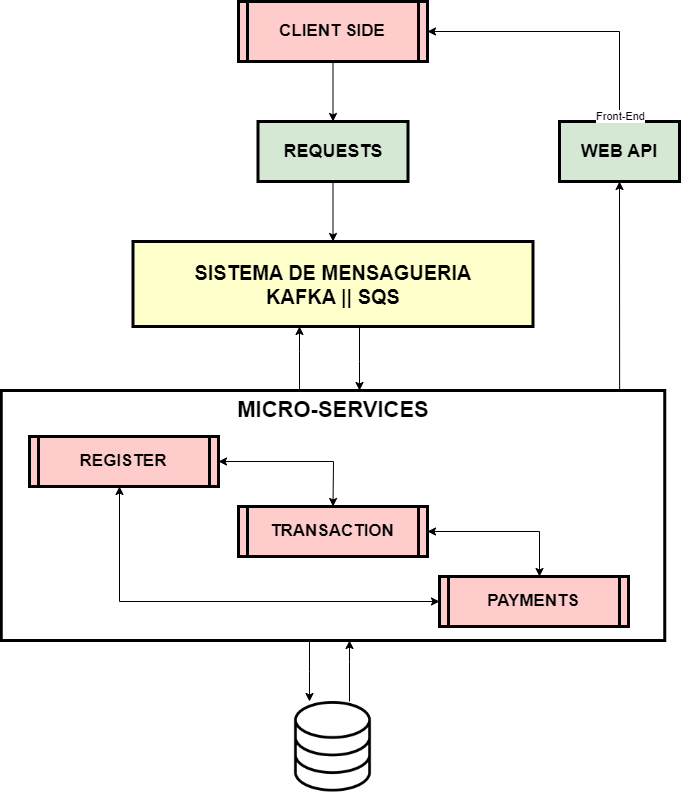
Quando tratamos da parte estrutural do software, fora proposto utilizar os conceitos de orientação a objetos, no entanto, chegou-se a conclusão de que utilizar a arquitetura orientada a eventos faria mais sentido para o modelo de sistema que seria desenvolvido, pois, a Event Driven Architecture (ou EDA) pode trazer benefícios como a escalabilidade do sistema, onde novos componentes podem ser adicionados sem afetar os existentes;

“Modelagem Dirigida a Eventos mostra como o sistema reage a eventos externos e internos. Ela é baseada na suposição de que um sistema tem um número finito de estados e que eventos podem causar uma transição de estado para outro.” (Sommerville, 2011. p 94)

A arquitetura de software orientada a eventos é uma alternativa à arquitetura orientada a objetos, e tem ganhado espaço no mundo da tecnologia. Em primeiro lugar, a arquitetura orientada a eventos permite uma maior flexibilidade na troca de informações entre os diferentes componentes do sistema. Em vez de depender de chamadas de métodos específicas, como acontece na arquitetura orientada a objetos, a comunicação entre os componentes é feita por meio da emissão e recepção de eventos. Isso faz com que o sistema seja mais modular e escalável, permitindo que novas funcionalidades sejam adicionadas sem grandes alterações na estrutura existente.

Além disso, este modelo é especialmente adequada para sistemas distribuídos, que precisam lidar com um grande volume de dados e processos simultâneos. Como cada componente do sistema é independente e pode responder a eventos de forma assíncrona, a arquitetura de eventos permite que o sistema seja mais tolerante a falhas e tenha uma melhor capacidade de recuperação. De forma geral, funciona da seguinte maneira: quando um evento ocorre em algum componente do sistema, ele emite uma mensagem que contém informações sobre o evento. Essa mensagem é então enviada a um ou mais componentes que estão interessados em receber esse tipo de evento. O componente receptor pode, então, executar alguma ação ou emitir um novo evento, dando continuidade ao fluxo do sistema.

### 3.2.1 Fluxograma do Sistema



### 3.2.2 Diagrama de Classes

### 

## 3.3 Casos de Uso

A modelagem de casos de uso é uma técnica importante em Engenharia de Software que ajuda a entender e documentar como um sistema deve se comportar em diferentes situações. Através da modelagem de casos de uso, é possível identificar os atores envolvidos no sistema, as suas interações com o sistema e as funcionalidades que o sistema deve prover. Essa técnica permite uma comunicação mais clara e precisa entre os desenvolvedores e o cliente, garantindo que as necessidades do cliente sejam atendidas e que o sistema seja desenvolvido de forma mais eficiente e eficaz.

“A modelagem da interação do usuário ajuda a identificar os requisitos do usuário. Destacando os problemas de comunicação que possam surgir, A modelagem de interação nos ajuda a compreender se a estrutura proposta para o sistema é suscetível de produzir o desempenho e a confiança requerida do sistema.” (Sommerville, 2011. p 86)

### 3.3.1 Modelo de Casos de Uso

**MCU – Saque:**

1. Usuário se identifica no sistema.

2. Software pergunta qual a quantia.

3. Usuário digita quantia.

4. Software pergunta se pode prosseguir.

5. Usuário prossegue.

6. Software pede sua senha de movimentação.

7. Usuário digita.

8. Software válida, gera uma ordem de saque e deixa pré-reservado a quantia a ser saca. Caso de uso termina.

**MCU – Transferência:**

1. Usuário se identifica no sistema.

2. Software pergunta qual tipo de transferência.

3. Usuário seleciona.

4. Software exibe um formulário de transferência.

5. Usuário preenche.

6. Software pergunta se está tudo correto e pode prosseguir.

7. Usuário prossegue.

8. Software pede a senha de movimentação.

9. Usuário digita.

10. Software válida, gera uma ordem de transferência e saldo é atualizado. Caso de uso se encerra.

**MCU – Depósito:**

1. Usuário se identifica no sistema.

2. Software exibe um formulário.

3. Usuário preenche.

4. Software pergunta se os dados estão corretos.

5. Usuário prossegue.

6. Software pede senha de movimentação.

7. Usuário digita.

8. Software válida, gera uma ordem de deposito e saldo é atualizado depois que for validado o deposito. Caso de uso finaliza.

**MCU – Cartões:**

1. Usuário se identifica no sistema.

2. Software exibe os cartões e a opção de solicitar um novo.

3. Usuário solicita novo cartão.

4. Software pergunta qual tipo.

5. Usuário seleciona.

6. Software exibe formulário de entrega.

7. Usuário preenche.

8. Software pergunta se os dados estão corretos.

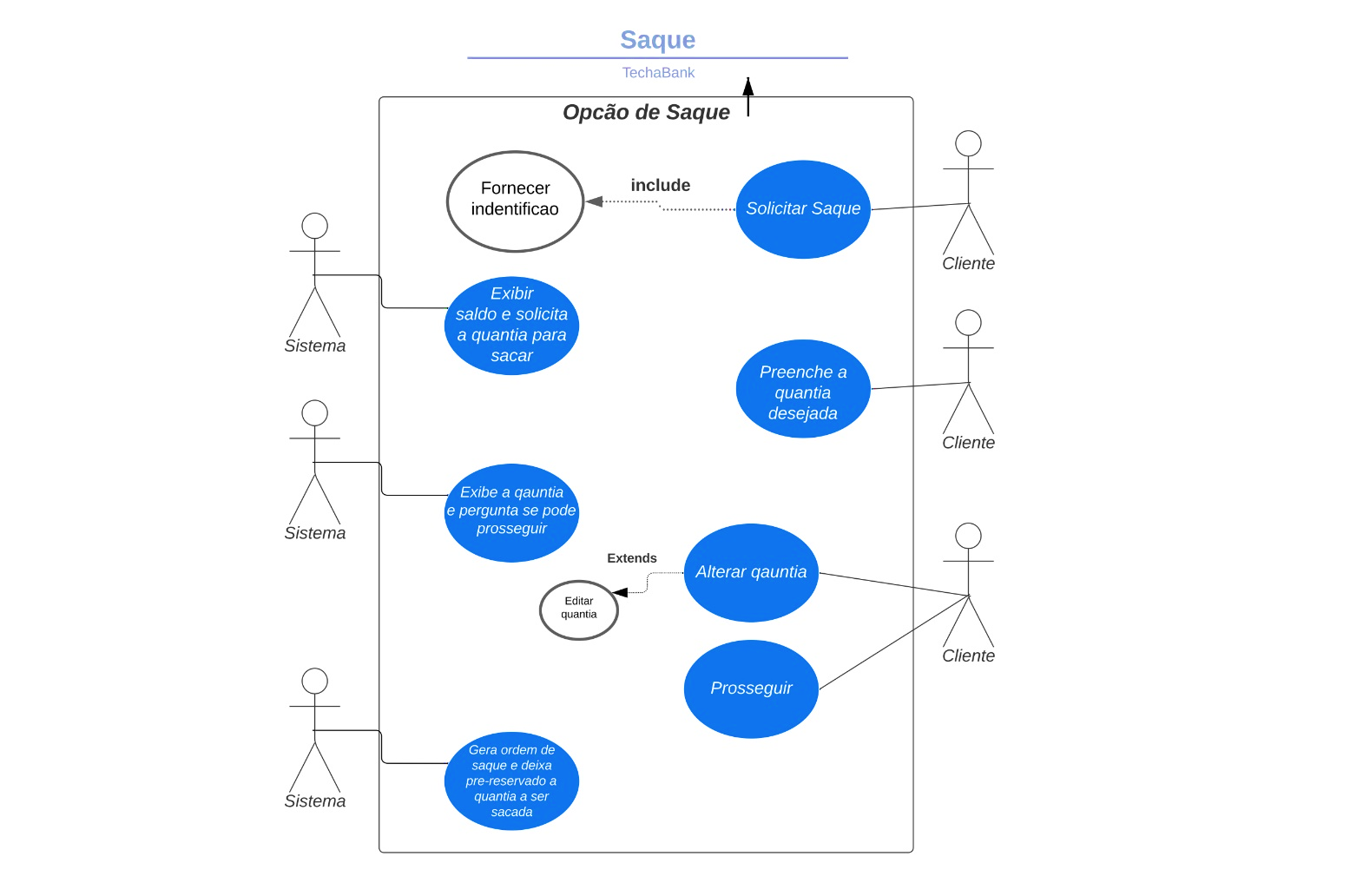
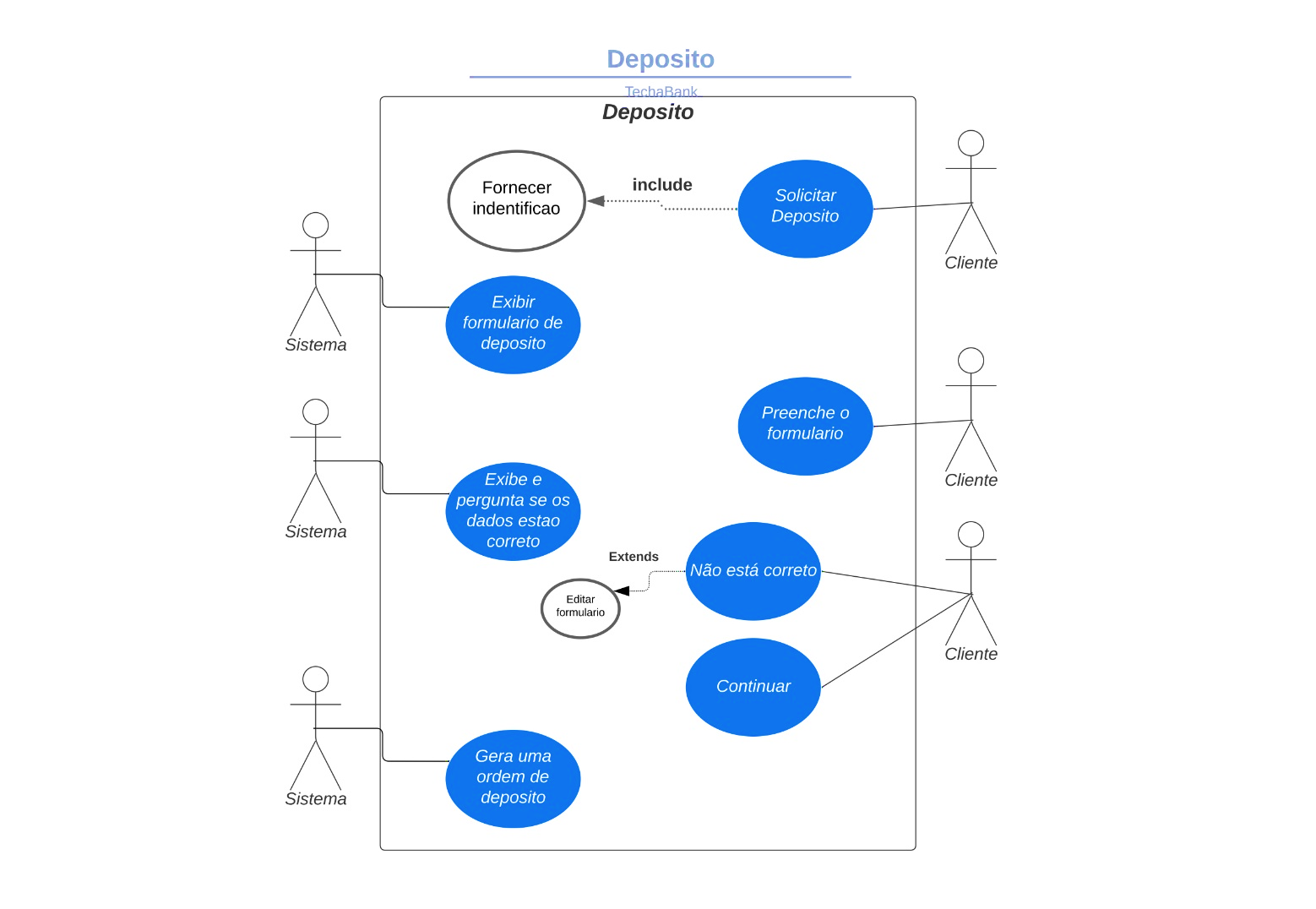
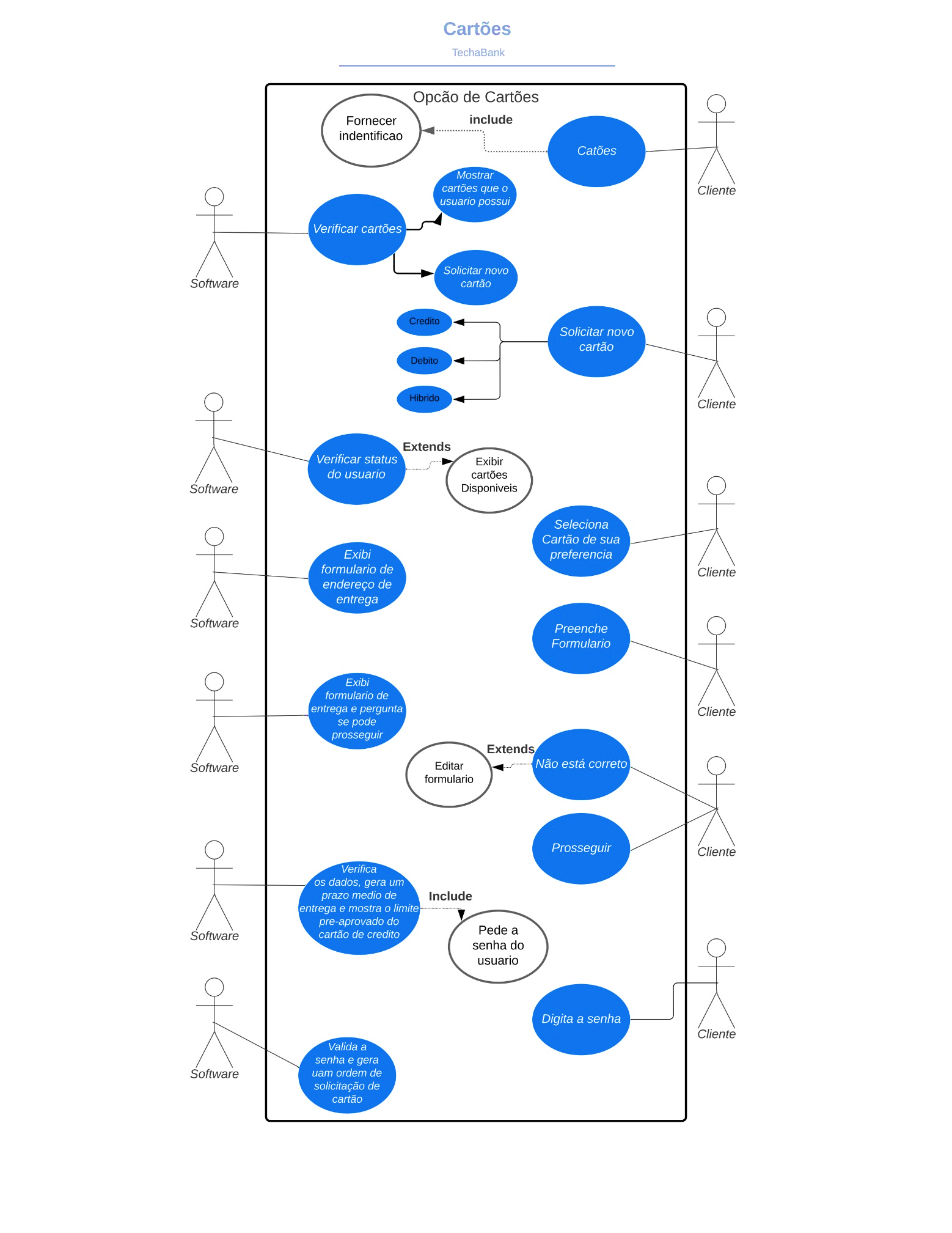
9. Usuário prossegue.

10. Software pede senha de movimentação.

11. Usuário digita.

12. Software gera uma ordem de solicitação de cartão, entrega um prazo médio de entrega e caso tenha função crédito é mostrado o saldo pré-aprovado. Caso de uso finalizado.

### 3.3.2 Diagrama de Casos de Uso



## 3.4 Modelo de Classes

### 3.4.1 Estrutura do Banco de Dados

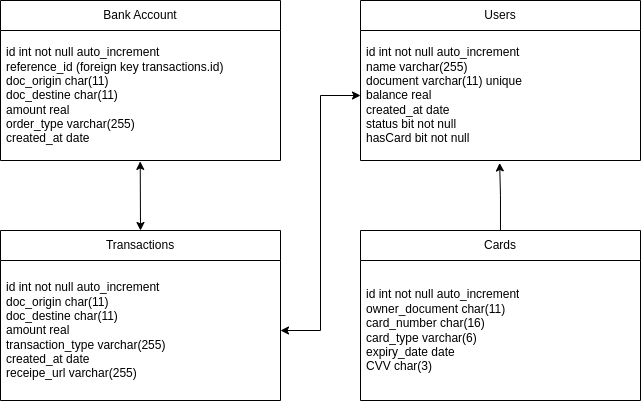
**Bank Account:** Recebe e faz transferências para outros bancos através de ordens que são feitas pelos usuários na plataforma. Ficando por parte do setor de tesouraria fazer a transação para a conta de destino se caso se tratar de um saque.

**User Account:** Possui cadastro de todos clientes cadastrados, assim como saldo de cada um. As transferências entre clientes do banco não passam pelo Bank Account, são feitas diretamente para a conta do outro usuário, alterando seu saldo, passando apenas por Transactions, sendo salvo no mesmo.

**Transactions:** Armazena todas transações que são feitas pelos usuários dentro da plataforma.

**Cards:** Armazena todos cartões dos clientes do Banco.

## 3.4.2 Diagrama de Fluxo de Dados



# 4 Considerações finais

## 4.1 Lições Aprendidas

* Colaboração é fundamental: A integração de diferentes especialidades demonstrou a importância da colaboração para obter resultados dignos de sucesso.
* Compreensão dos requisitos do usuário: A análise detalhada dos requisitos dos usuários é essencial para o sucesso do projeto.
* Abordagem sistemática: Seguir uma abordagem sistemática, dividindo o projeto em etapas claras, ajuda a garantir a qualidade e a minimizar erros.
* Modelagem efetiva: A modelagem adequada, por meio de diagramas e modelos, facilita a comunicação e a identificação de problemas antes da implementação. Boas práticas de engenharia de software: A aplicação de boas práticas, como modularização, reutilização de código e testes, melhora a eficiência e a qualidade do software.
* Flexibilidade e adaptação: Lidar com desafios inesperados requer flexibilidade e habilidade para encontrar soluções criativas.

## 4.2 Conclusão

A priori, a ideia do projeto era aplicar os ensinamentos das matérias de Engenharia de Software, tanto como Análise e Desenvolvimento de Projeto, dentro de um pequeno modelo de um banco digital, satisfazendo assim os requisitos necessários e demandas da empresa fictícia criada no início de tudo, construindo um modelo em código que simulasse as ações do usuário, tornando nítido a semelhança do mesmo com qualquer outro serviço bancário tendo como foco as competências ensinadas em sala.

Todavia, durante o desenvolvimento do mesmo, percebeu-se o quão importante se fazem os processos que permeiam todo o decorrer do desenvolvimento do projeto, desde a concepção até a finalização, por exemplo: definição dos requisitos e regras de negócio; engajamento de equipe; idealização do projeto; objetivo a ser alcançado e entre outros. Foi possível perceber o impacto que se possui no que diz respeito a metodologia escolhida, pois, um projeto que se comporta de maneira mais fluída e sem parâmetros de desenvolvimento definidos (KPI) é facilmente procrastinável. A flexibilidade não pode ser colocada jamais na posição de aceitar algo fora do padrão de qualidade esperado, devendo se ter o prazo de entrega como parâmetro de medidas a serem tomadas no caso de se perceber uma deficiência por parte de qualquer integrante em relação as entregas planejadas de acordo com a EAP.

Em especial, fora percebido uma falha na comunicação entre as partes da equipe, onde ocorreu uma distorção no que imaginava-se ter como resultado final, que seria um código que possui-se as funções necessárias para alteração e persistência em um banco de dados (CRUD), entregando algo que fosse mais próximo de uma API.

Por fim, acredita-se que aquilo que foi aprendido durante o desenvolvimento do projeto como um todo vale mais do que qualquer perfeccionismo exigido pelo próprio grupo em si. Pode se dizer que sobretudo, desconsiderando os problemas que surgiram, o trabalho em si fora feito atendendo a proposta que fora sugerida pelos professores, sendo possível ter noção da importância dos conteúdos ministrados em sala.

# 5 REFERêNCIAS

* SOMMERVILLE, Ian. **ENGENHARIA DE SOFTWARE**. 9. ed. SÃO PAULO: PEARSON, 2011.
* MIT BOOTCAMP: **BUILDING A BANK FROM SCRATCH**

<https://www.youtube.com/watch?v=Hpv__uhbT6k>” acessado em 23/04/2023

* CANAL **DEV FULL CYCLE**

<https://www.youtube.com/@FullCycle> acessado em 16/04/2023

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint-log (Board de Tarefas)** | | | | | | | | | |
| **INTEGRANTES** | | **FUNÇÕES** |  |  | |  | |  | |
| Derek Gonçalves | | Analista de Sistema |  |  | |  | |  | |
| Gabriel Oliveira | | Desenvolvedor Pleno |  |  | |  | |  | |
| Geovani Monteiro | | Desenvolvedor Sênior |  |  | |  | |  | |
| Kainan Guerra | | Product Owner |  |  | |  | |  | |
| Kevin Nunes | | Analista de Negócios |  |  | |  | |  | |
| **TAREFA** | **TÍTULO** | | | | **DIFICULDADE** | | **PRIORIDADE** | | **RESPONSÁVEL** |
| TPI-01 | Definir e digitar as Regras de negócios | | | | 6 | | Alta | | Kainan |
| TPI-02 | Desenvolver Plano de Negócios | | | | 2 | | Baixa | | Kevin |
| TPI-03 | Desenvolver o código de simulação do sistema (Front-End) | | | | 9 | | Alta | | Geovani e Gabriel |
| TPI-04 | Desenvolver as artes gráficas que serão utilizadas na demonstração | | | | 2 | | Alta | | Geovani |
| TPI-05 | Iniciar a modelagem da parte escrita do projeto | | | | 3 | | Alta | | Kainan |
| TPI-06 | Desenvolver Telas de Mock-up de Ação do Usuário no sistema | | | | 4 | | Alta | | Geovani |
| TPI-07 | Pesquisar por referências bibliográficas | | | | 8 | | Média | | Kainan |
| TPI-08 | Redigir Estrutura do Banco de Dados | | | | 7 | | Alta | | Kainan |
| TPI-09 | Desenvolver modelo de apresentação do projeto | | | | 8 | | Média | | Derek |
| TPI-10 | Fazer Backlog do projeto | | | | 9 | | Média | | Kainan |
| TPI-11 | Desenvolver UML/fluxograma de Casos de Uso (MCU/DCU) | | | | 3 | | Baixa | | Kevin |
| TPI-12 | Desenvolver fluxograma de fluxo de dados | | | | 2 | | Média | | Kainan |
| TPI-13 | Redigir ciclo de vida e metodologia do projeto | | | | 4 | | Alta | | Kainan |
| TPI-14 | Fazer Análise SWOT do Sistema | | | | 5 | | Alta | | Kevin |
| TPI-15 | Definir Requisitos Funcionais e Não Funcionais | | | | 5 | | Alta | | Kainan |
| TPI-16 | Escrever as Lições Aprendidas e Conclusão do Projeto | | | | 7 | | Baixa | | Derek |
| TPI-17 | Desenvolver Fluxograma do Sistema com Arquitetura Aplicada | | | | 10 | | Alta | | Kainan |
| TPI-18 | Desenvolver EAP (Estrutura Analítica do Projeto) | | | | 5 | | Baixa | | Kainan |
| TPI-19 | Fazer Diagrama de Classes de Objetos | | | | 8 | | Média | | Kevin |
| TPI-20 | Redigir Metodologia e Estrutura do Projeto | | | | 6 | | Média | | Kainan |
| TPI-21 | Desenvolver parte de Autenticação do Usuário | | | | 8 | | Alta | | Geovani |